



# NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

## **Approaches for Measuring the Management Effectiveness of Software Projects**

by

Kadir Alpaslan Demir, James Bret Michael, and John S. Osmundson

April 2008

**Approved for public release; distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

**NAVAL POSTGRADUATE SCHOOL**  
**Monterey, California 93943-5000**

Daniel T. Oliver  
President

Leonard A. Ferrari  
Executive Vice President and Provost

This report was prepared for the Computer Science Department.

Reproduction of all or part of this report is authorized.

This report was prepared by:

---

Kadir Alpaslan Demir  
PhD Candidate in Software Engineering

---

James Bret Michael  
Professor of Computer Science

---

John S. Osmundson  
Research Assoc. Professor of  
Information Science

Reviewed by:

Released By:

---

Peter J. Denning  
Chairman  
Department of Computer Science

---

Dan C. Boger  
Interim Vice President and  
Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> April 2008	<b>3. REPORT TYPE AND DATES COVERED</b> Technical Report	
<b>4. TITLE AND SUBTITLE:</b> Title (Mix case letters) Approaches for Measuring the Management Effectiveness of Software Projects			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Kadir Alpaslan Demir, James Bret Michael, and John S. Osmundson				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b>  Effective software project management is a key element in achieving software project success. In order to improve the quality of the management and focus our efforts on the right issues, it is essential to measure software project management effectiveness first. In this report, we introduce four alternative approaches for guiding the development of project management metrics.				
<b>14. SUBJECT TERMS</b>  Software Project Management, Software Metrics, Measurement of Management Effectiveness of Software Projects			<b>15. NUMBER OF PAGES</b> 35	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UU	

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

Effective software project management is a key element in achieving software project success. In order to improve the quality of the management and focus our efforts on the right issues, it is essential to measure software project management effectiveness first. In this report, we introduce four alternative approaches for guiding the development of project management metrics.

THIS PAGE INTENTIONALLY LEFT BLANK



## TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	SUCCESS PYRAMID .....	3
III.	DISCUSSION OF APPROACHES.....	7
	A. SUBJECTIVE EVALUATION .....	7
	B. QUESTIONNAIRE-BASED MEASUREMENT .....	8
	C. METRICS-BASED MEASUREMENT .....	9
	D. MODEL-BASED MEASUREMENT.....	11
IV.	CONCLUSIONS .....	15
	LIST OF REFERENCES.....	17
	INITIAL DISTRIBUTION LIST .....	23

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

Figure 1.	Success Pyramid .....	5
Figure 2.	The Scope of Success at Different Levels .....	6
Figure 3.	Four Approaches for Software Project Management Effectiveness Measurement.....	7

THIS PAGE INTENTIONALLY LEFT BLANK

## I. INTRODUCTION

There are various studies reporting the success and failure rates of software projects [GA,CH,EL]. Even with the lowest failure rates reported, the software projects are significantly failing when compared to projects in other fields. In [SL], current project management issues in leading project-based industries are listed. Among nine industries, in only software industry column, overruns and poor performance is explicitly listed as an issue among others. The average software project is likely to be six to 12 months behind schedule and 50 to 100 percent over budget [YO]. One would expect that our record in software projects should have been much better with all the advancements in technical aspects of software engineering. However, we believe relying merely on technological advances would be misleading. We also need significant advances in software project management field to achieve better results in software projects. Therefore, proposals and discussions for applicable and viable theories, models, tools and practices in software project management are important steps in achieving better project outcomes.

Ineffective software project management is among the main reasons for the failures in software projects [JO]. In addition, effective project management is a determinant in the success of the software projects [JO]. DeMarco and Lister state that “For overwhelming majority of the bankrupt projects we studied, there was not a single technological issue to explain the failure.” [DE]. Robertson et. al. emphasize that “In several decades of project experience, we have never seen a project fail for technical reasons. It has always been human failures that have caused otherwise good projects grind to a halt.” [RO]. Various other studies, researchers and practitioners report similar issues regarding the importance of software project management in the success and failure of software projects [WE, DS].

According to Boehm, poor management can increase software costs more rapidly than any other factor. COCOMO [BO1], a method for software project cost and effort estimation developed by Barry Boehm and his colleagues, does not include project management as a factor. Therefore, in COCOMO II [BO2], the estimation model incorporates some project management related factors such as PCON (personnel

continuity) and PMAT (process maturity). We believe, in order to keep the rate of the software cost overruns and schedule slippages down, measuring and therefore improving the quality of project management areas is an enabler. In addition, such project management metrics can be incorporated to cost estimation techniques yielding better estimates.

According to Morris, “One of the major areas of project management development over the next years, I believe, will be establishing and refining inter-industry metrics for quantifying performance improvements. Much of this work will be IT-related.” [MO]. Hyvari investigates the effectiveness of project management based on four different factors [HY]. The factors are organizational structures, technical competency, leadership ability, and the characteristics of an effective project manager. He does not state the reasoning for selection of these factors and whether this is a complete list or not.

Project management is a complex endeavor and development of a metric for project management effectiveness is clearly not an easy task. However, measurement and evaluation of management effectiveness in software projects opens up a lot of opportunities for improvement. In this report, we introduce four approaches for measuring the quality of software project management. We further discuss each approach and present examples of the existing implementations. The significance of the report is the guidance for the development of project management effectiveness metrics.

## II. SUCCESS PYRAMID

Project management success is not the same as project success [CO2]. Even though most practitioners would emphasize that software project success is closely related to project management quality or success, there is no established scientific evidence for such relation in the software project management literature. Related empirical studies in the software engineering field or even in the project management literature are quite limited. This is no coincidence. There are some reasons:

1. Even though there are many studies in the area of project success factors, there is no well-established criteria for project success. Pinto and Slevin state that words like success and failure are in the eyes of the beholder. They also emphasize the risk of mislabeling projects as success instead of failure or vice versa without a well-established set of project success criteria [PI]. For example, Proccacino investigated how various practitioners view project success. His study adds and introduces another view to existing project success criteria [PR1]. White criticizes the lack of suitable measures of successful projects [WH]. Simply, we still don't have a universally-accepted definition for project success. Then, how can we relate project success to project management success when there is no clear definition for project success?
2. There is no theory for project management that has found recognition [SM, JU]. In 2006, Turner, editor of the International Journal of Project Management wrote a series of editorials. In these editorials, he states that project management has still not been accepted as an academic discipline [TU1]. He concludes that one of the reasons for that is the lack of a theory for project management. In that and following editorials, he provides a normative theory of project management [TU1,TU2,TU3]. In 2007, Sauer and Reich wrote a response. While they promote the idea of having a normative theory for project management, they expressed the need for a theory that helps us to understand the conditions, constraints, and drivers leading to functional and dysfunctional behaviors [SA]. Therefore, we can influence such behavior to reach intended results. While theories shape a

discipline, they also guide researchers to investigate the phenomenon. As a result, our ability to develop quality criteria for project management is limited.

3. The fields of software engineering and project management are quite young when compared to other fields. Research works related to foundations of disciplines take time to build up. Reliable empirical studies require the existence of a certain amount of fundamental research. Therefore, our ability to conduct empirical research in the field of software project management is limited.

Defining project success is not an easy task. It is multifaceted and difficult to measure [GR]. The three conventional project success criteria are time, cost, and performance. Pinto and Rouhiainen state that these criteria don't work in the modern business world [PI2]. The tremendous competition in this modern business world requires a customer-oriented focus. Therefore, customer satisfaction is another key criterion. Glass points out the need for a new theory of project success [GL]. Different stakeholders may have different concerns. This is inevitable. One of the key challenges of any project management is to align the goals and addressing the concerns of the stakeholders. Linberg showed that the definition of success for software practitioners is quite different from the conventional criteria [LI]. Software practitioners may classify a project as success even though it is late, over budget. They are more concerned with the quality and functionality of the product. In addition, they may even view a cancelled project as success due to the lessons learned and the challenge in the project. Agarwal and Rathod investigated the notion of software project success for different stakeholders [AG]. They examined project success in the views of programmers/developers, project managers, customer account managers. Procaccino developed a quantitative model for early assessment of software development success in the practitioner's perspective [PR2, PR3]. Cooke-Davies examines the issue with a broader view [CO1,CO2]. His view clarifies some challenged research areas beautifully. He provides a definition of success at different levels. His questions for each level help us to focus to the big picture. According to Cooke-Davies, there are three levels of success:

*Level 1: Project Management Success: Was the project done right?*



*Level 2: Project Success: Was the right project done?*

*Level 3: Consistent Project Success: Were the right projects done right, time after time?*

These levels are shown in a pyramid in Figure 2. The figure implicitly implies that the success of each level depends on the success of the previous level. Even though, this is the fact in many cases, not in all cases. The figure has the merit of providing an overall view of what success means at each level. It is possible to achieve a successful project even when the management fails or vice versa [MU]. For example, even though the management has done a good job in completing the project within budget, on time and with the expected quality, the product may never find its share in a competitive market. Then, the fault lies on the executive management (or project sponsor) with the decision to undertake such a project delivering a product that cannot find its place in a competitive market. In that case, the assumption is that the project management team is handed the project proposal and they are to deliver project.



Figure 1. Success Pyramid

Munns and Bjeremi provide a good discussion regarding the role of project management in achieving project success [MU]. Munns and Bjeremi discuss that project management success suggests a shorter term while project success has a longer term. This

is consistent with Cookie-Davies's view of success at different levels. As a result, the developed framework for success at different levels is presented in Figure 2.

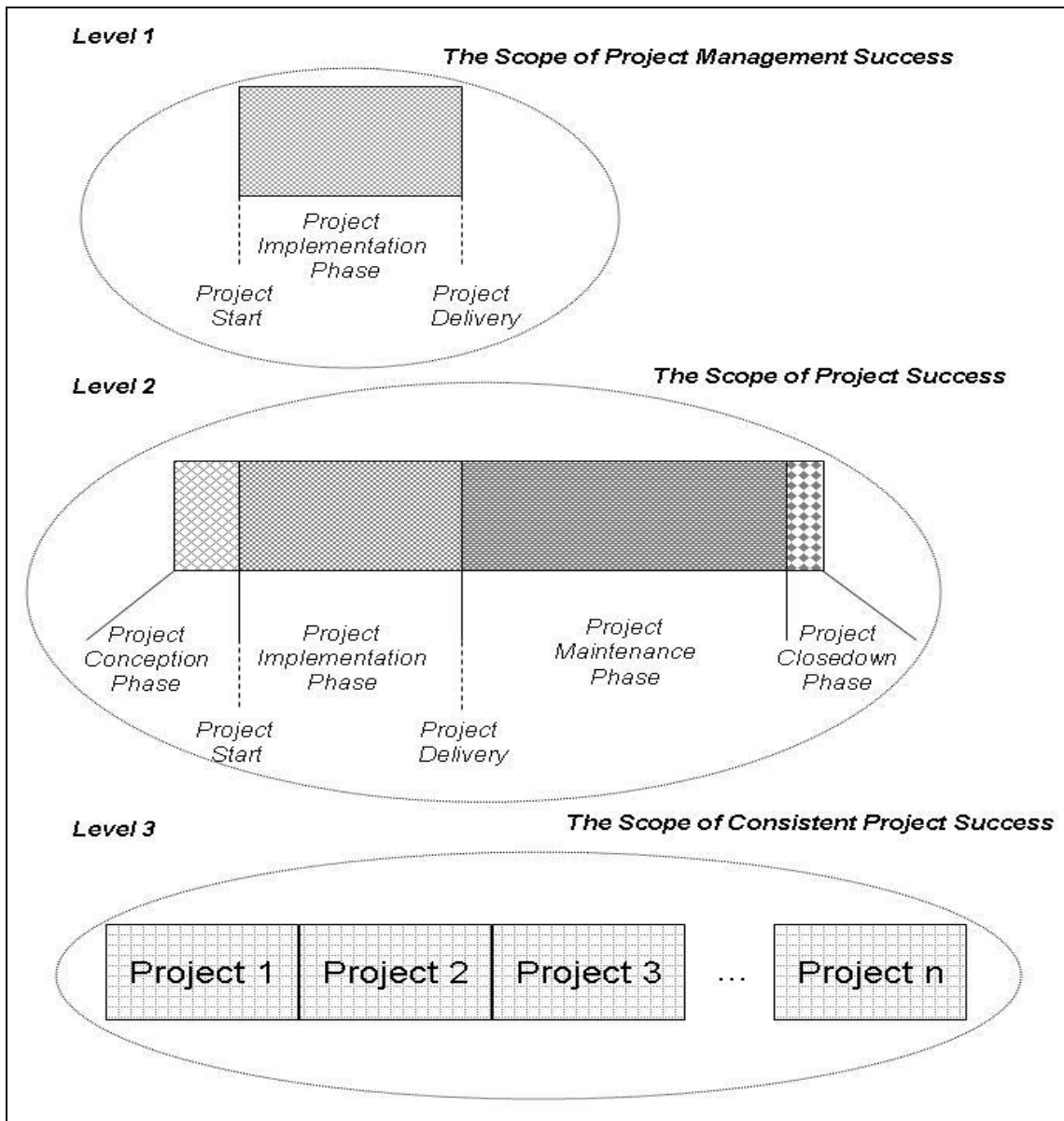


Figure 2. The Scope of Success at Different Levels

### III. DISCUSSION OF APPROACHES

To our knowledge, this study is the first attempt to provide a framework for measuring the effectiveness of software project management. Related measurement studies in the project management literature are almost non-existent. The management literature focuses on organizational effectiveness that is only remotely related to project management effectiveness.

We have identified four different approaches that can be used in the development of methods to measure the effectiveness of software project management. Figure 3 shows these four approaches and corresponding metric types. Each of these approaches is discussed in the following sections.

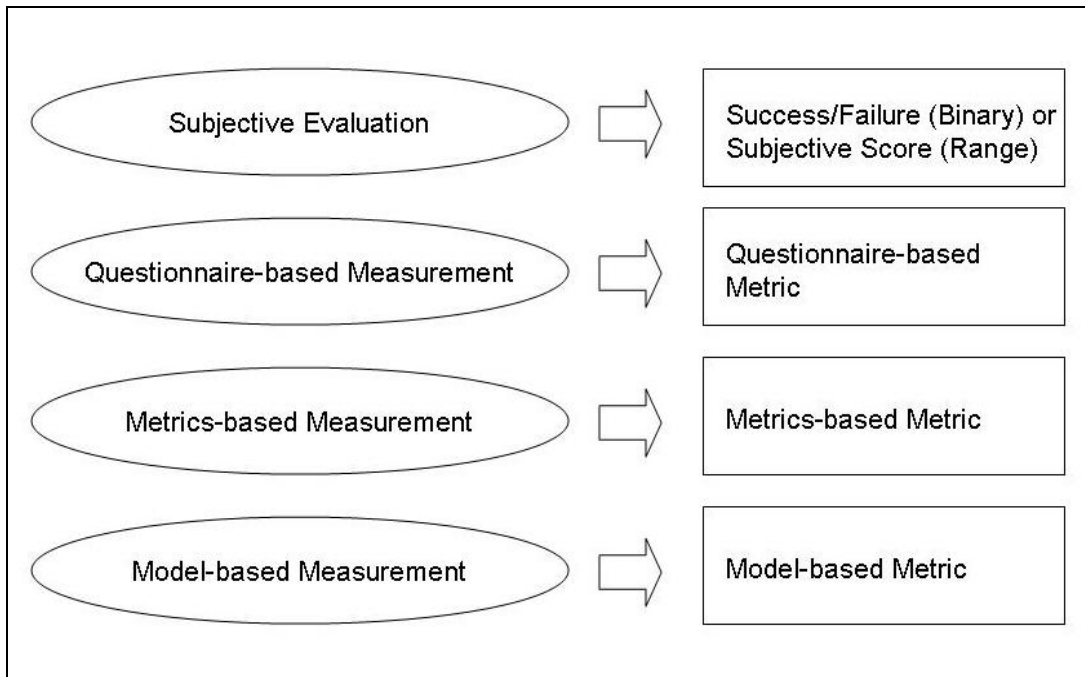


Figure 3. Four Approaches for Software Project Management Effectiveness Measurement

#### A. SUBJECTIVE EVALUATION

In this approach, the project participant's perception is used in the evaluation of the project management. This participant may be the project manager, the technical manager, or the developers. Since it is based on the perception of the participant, this is a

subjective evaluation. In this approach, the project participant is simply asked to categorize the project as a success/failure or rate the project based on a scale. This approach is the simplest one and used in some studies. For example, Osmundson et. al. [OS] requested the project managers and project developers rate the project's success based on a scale from 0 to 10 in their study. In another study, Verner and Evanco investigated the project management practices leading to success in in-house software development [VE]. They analyzed 42 successful and unsuccessful projects based on the senior software practitioners' categorization of their projects. In his doctoral dissertation, Procaccino used the same approach and his study is based on the view of software practitioners [PR2].

It is important to point out that even though such approach is subjective; it is hard to disregard the validity (to some extent) of the project participant's perception. The practitioners have a sense of what the best practices are and if those are followed or not. However, as Pinto and Slevin [PI1] pointed out there is a significant risk of mislabeling a project as a success or failure without a well-established set of success criteria. This risk is more significant when the study compares the successful and failure projects based on the subjective evaluation approach. Because when the project is in fact a failure and the participant mislabels it as a success, then this evaluation skews both results such as boosting the success rate and decreasing the failure rate.

Another important consideration is that the measures resulting from this approach do not provide any insight on how to improve the management of the project. Just labeling a software project as a success or a failure without understanding the causes of it, has limited use for practitioners and researchers.

## **B. QUESTIONNAIRE-BASED MEASUREMENT**

In this approach, the measurement of management effectiveness is based on the evaluation of responses to a questionnaire. Questionnaire-based evaluations are common in management and organizational sociology study areas (for example [BR, BA, PA, KI, MU]). Because abstract concepts such as teamwork, organizational commitment,

communication, leadership etc. are hard to quantitatively analyze. This approach has been used in the development of a quality management metric for software development [OS].

In the study by Osmundson et. al., a questionnaire was developed to investigate which best management practices are followed to what extent in a software project. Then, based on the responses to the questions, the quality of the project management is measured. They also compared the resulting metric (QMM) with a metric gathered via subjective evaluation discussed in the previous section. The questionnaire investigates four important areas of software project management. They are requirements management, project planning and estimation, risk management and people management [MN]. People management is further divided into four areas: Human resources, leadership, communication, technical competency of the program manager. The complete questionnaire instrument included 457 questions. The QMM metric is based on a scale from 0 to 10, 0 being the lowest quality score, and 10 being the highest quality score. The importance of the QMM study is the focus on the development of a metric for the quality or effectiveness of project management in software projects.

COCOMO II incorporates a process maturity factor (PMAT) as a scale factor to the effort estimate [BO2]. It is important to note that scale factors affect the effort estimate exponentially. In COCOMO II, this PMAT factor is determined using one of two methods [CL]. The first method is based on the SW-CMM rating of the organization when there is one. The second method is used when the organization does not have a SW-CMM rating. The second method uses another rating (Equivalent Process Maturity Level – EPML) which is based on the percentage of compliance for each key process area goal in SW-CMM model. This compliance is (EPML rating) evaluated via the responses to a questionnaire derived from 18 key process areas.

### **C. METRICS-BASED MEASUREMENT**

Another approach for measuring the effectiveness of software project management is via the use of other software metrics. For example, metrics such as the number of defects over time, software complexity, requirements stability, staff turnover

rate etc. can be used as inputs for a metric model for software project management effectiveness metric. This type of measurement is in fact an indirect measurement. When complex attributes are measured in terms of simpler sub-attributes, this measurement is indirect [FE]. Many effort predictions use several levels of indirect measurement [FE]. Erdogmus presents a cost-effectiveness indicator for software development. He uses base measures such as nominal output, production effort, rework effort, issue count, staffing profile to derive a breakeven multiple as an indicator aggregating productivity, quality, and staffing needs[ER]. This is a good example for this approach in a different context. Wohlin and Maryhauser provide a detailed method for assessing software project success using subjective evaluation factors [WO].

To our knowledge, there has not been an attempt for the development of a metric for assessing the management effectiveness of software projects using this approach. Therefore, we provide a metric model for such measurement to guide the future researches. The model is shown below:

$$SPMEM = Measurement\_function(\sum_{i=1}^n w_i m_i)$$

In the model above,  $m$  is a metric that has found to relate to the metric for management quality, which is denoted by  $SPMEM$ . There can be  $n$  number of metrics. There may also be only one metric and in that case  $n$  equals to 1. Examples of such metrics may include programmer productivity, defect reduction rate, certain earned value metrics (EVM) metrics etc.  $w_i$  is the weight associated with a certain metric,  $m_i$ . Such weights may be required since different metrics may relate to the resulting management quality metric differently. Then these metrics are combined via a measurement function depending on the hypothesized metric model.

Above we presented a generalized metric model. Development of a management effectiveness or quality metric for software projects using this approach requires significant research based on empirical studies.

#### **D. MODEL-BASED MEASUREMENT**

In this approach, the metrics for effectiveness or quality of the management are derived from models of management of software projects. Currently, this approach is also conceptual and there are no examples implemented. There has not been any attempt to measure the management effectiveness of software projects based on a model of project management.

For quite some time, researchers are focused on developing software development life-cycle methodologies. There are many examples of methodologies such as waterfall, spiral, win-win, rapid prototyping, agile development, SCRUM etc. There is also a field called software process research within the software engineering discipline. Software process research started back in 1980's through a series of workshops and events. Due to many software application failures, researches focused on improving the software process. The assumption is that there is a direct correlation between the quality of the software process and the quality of the software application developed. A good example in the software process research is the development of the CMM series models. An area of software process research is software process modeling. There are a number of Process Modeling Languages (PMLs) developed [FU]. Some examples are Process Interchange Format (PIF) [GR1,GR2], Process Specification Language (PSL) [SC], Unified Process Model (UPM) [KR], Core Plan Representation (CPR) [PE], Workflow Management Coalition Process Definition (WfMC) [WfMC], Architecture of Integrated Information Systems (ARIS) [SCH]. A review of these PMLs can be found in [BR]. In June 2005, Business Process Management Initiative (BPMI) and Object Management Group OMG merged their activities and formed the Business Modeling & Integration (BMI) Domain Task Force (DTF). They have developed various standard proposals for different views of process management such as Business Motivation Model (BMM) specification [BMM], Business Process Definition Metamodel (BPDM) [BPDM]. Even Gantt Charts and PERT (Program/Project Evaluation and Review Technique) and CPM (Critical Path Analysis) charts are process models and development of Gantt Charts dates back to 1910s. However, there is a significant difference between the PMLs mentioned above and the process models. While the process models (such as Gantt, PERT, CPM) got wide-

acceptance in industry, as Fuggetta [FU] pointed out few (if any) of the proposed PMLs and related Process-centered Software Engineering Environments (PSEE) have been transferred into industrial practice. Fuggetta states that the goal should be to ease the adoption of PMLs. Most of the PMLs are heavily technical and formal. The wide adoption of Gantt, PERT and CPM charts tell us what the practitioners would like to see in these types of process modeling languages: It is simplicity. Since these PMLs could not find their share in practicality, we do not have actual project data based on models developed with these languages. Viable effectiveness measurements for software project management require actual data from projects, which we do not have. Process models are developed for one specific purpose and they only focus one aspect of the project management. For example, PERT charts are used for prediction of the project schedule. However, managing software projects has many aspects.

As a result, Pinto stresses the importance of modeling the business, technical, financial, environmental, and other dimensions of the project before committing any significant sources or even before the go-ahead [PI3]. Jaafari provides a simplified highest-level representation of a project model and lists the ideal requirements for a project model [JA]. He stresses that we still have a long way to go in realizing such sophisticated modeling systems. We have developed a simple, visual and formal modeling language called PROMOL for modeling project management [DM]. This modeling tool achieves most of the ideal requirements listed by Jaafari. According to Demir and Osmundson, as hypothesized in [DM], there are two core concepts in the heart of project management. They are activities and entities. These two concepts can be used in modeling project managements. Then, the quality or effectiveness of these activities and entities in a project management model can be used as inputs for a metric model for effectiveness of project management. As a result, a high-level metric model may be formulated as follows:

$$SPMEM = Measurement\_function(\sum_{i=1}^m qa_i + \sum_{j=1}^n qe_j)$$



In the metric model above,  $qa_i$  is the quality of an activity and  $qe_i$  is the quality of an entity. These activities and entities are components of a project management model. There can be  $m$  number of activities and  $n$  number of entities in the model that is of interest as inputs for the *SPMEM* metric model. The measurement function is a function that combines the quality measures of activities and entities. This function is specific to the metric model and it is defined in the metric model. Different metric models may require quite different measurement functions. It is important to emphasize that there can be a number of variations of this high-level model. Examples of these variations may be including only activities, or including only entities or basing the metric model to a specific life-cycle development model and deriving the activities and entities from this life-cycle development model.

The success of the model-based measurement will be highly dependent on the representation capability of the project management model. When these project management models are far from satisfactory, then the resulting metric will likely be unsatisfactory.

THIS PAGE INTENTIONALLY LEFT BLANK

## IV. CONCLUSIONS

According to Evans, Abela and Beltz, the first characteristics of dysfunctional software projects is failure to apply essential project management practices [EV]. This is derived from 841 risk events in 280 software projects. 480 out of 841 risk events (57%) in software projects are due to not applying essential project management practices. Jones reports that an analysis of 250 software projects between 1995 and 2004 reveals six major areas effective in successful projects and inadequate in failing projects [JO]. They are project planning, project cost estimating, project measurements, project milestone tracking, project change management, project quality control. All of these areas are related to software project management. These studies clearly show the importance of project management in achieving software project success. Therefore, project management metrics are the keys to rationally focus and substantiate the management improvement efforts.

It is important to note the recognized work by Basili and Rombaugh on the Goal/Question/Metric (mostly known as GQM approach) approach for development of software metrics [BA]. They provide an overall approach on how to develop metrics. First, it is very important to define the goal of the measurement activity. This sets up the context for the measurement. Second, we have to find the right questions for identifying the metrics that are going to be used in the measurement effort. Third, we have to choose or develop the right metrics for achieving the goal. The GQM approach is completely applicable to all the approaches presented here. The goal referred in GQM is already defined via the context and it is measuring the quality or effectiveness of management of a software project. The four approaches help us to refine and ask the right questions. The examples and high-level models presented in the previous sections guide us in identifying and combining the necessary metrics.

In this report, we have achieved:

- A good review of the literature regarding the effectiveness measurement of project management

- The introduction of four approaches for effectiveness measurement for software project management
- The guidance for the development of project management metrics via high-level metric models

## LIST OF REFERENCES

[JO] C. Jones, “Software Project Management Practices: Failure versus Success”, *Crosstalk*, Vol. 17, No. 10, October 2004.

[GA] Contracting for computer software development, FGMSD-80.4, US General Accounting Office, Washington, DC, 1979.

[CH] CHAOS, in: The Standish Group Report, Standish Group, West Yarmouth, MA, 1995

[EL] K. E. Emam, and G. Koru, “A Replicated Survey of IT Software Project Failure Rates”, *IEEE Software*, 2008, accepted to appear.

[DE] T. DeMarco, and T. Lister, *Peopleware: Productive Projects and Teams*, 2nd Edition, Dorset House Publishing Company, New York, NY, 1999.

[RO] S. Robertson, and J. Robertson, *Requirements-Led Project Management*, Pearson Education Inc., Boston, MA, 2005.

[WE] G. Weinberg, *Quality Software Management: Volume 3 Congruent Action*, Dorset House, New York, 1994.

[BO1] B. W. Boehm, *Software Engineering Economics*, Prentice-Hall, Upper Saddle River, NJ, 1981.

[BO2] B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, *Software Cost Estimation with COCOMO II*, Prentice-Hall PTR, July 2000.

[DS] Defense Science Board, “Report of the Defense Science Board Task Force on Defense Software”, Office of the Under Secretary of Defense for Acquisition and Technology, Washington, D.C. 20301-3140,  
<http://www.acq.osd.mil/dsb/reports/defensesoftware.pdf>, accessed in March 2008.

[MO] P.W. G. Morris, *Chapter 1 in Project Management Handbook*, Editor J. K. Pinto, Project Management Institute, Jossey-Bass Inc., San Francisco, California, 1998, p. 23.

[HY] I. Hyvari, "Project management effectiveness in project-oriented business organizations", *Int. Journal of Project Management*, Vol. 24, 2006, pp. 216-225.

[AG] N. Agarwal, and U. Rathod, "Defining "success" for software projects: An exploratory revelation", *Int. Journal of Project Management*, Vol. 24, 2006, pp. 358-370.

[PI1] J.K. Pinto, *Project Management Handbook*, Project Management Institute, Jossey-Bass Inc., San Francisco, California, 1998, p. 379.

[SM] H. J. Smyth, and P. W.G. Morris, "An epistemological evaluation of research into projects and their management: Methodological issues", *Int. Journal of Project Management*, Vol. 25, 2007, pp. 423-436.

[JU] J. Pollack, "The changing paradigms of project management", *Int. Journal of Project Management*, Vol. 25, 2007, pp. 266-274.

[TU1] J. R. Turner, "Towards a theory of project management: The nature of the project", *International Journal of Project Management*, Vol. 24, 2006, pp. 1-3.

[TU2] J. Rodney Turner, "Towards a theory of project management: The nature of the project governance and project management", *International Journal of Project Management*, Vol. 24, 2006, pp. 93-95.

[TU3] J. Rodney Turner, "Towards a theory of project management: The function of project management", *International Journal of Project Management*, Vol. 24, 2006, pp. 187-189.

[SA] C. Sauer, and B.H. Reich, "What do we want from a theory of project management? A response to Rodney Turner", *International Journal of Project Management*, Vol. 25, 2007, pp. 1-2.

[GR] A. Griffin, A. L. Page, "PDMA Success Measurement Project: Recommended Measures for Product Development Success and Failure", *Journal of Product Innovation Management*, Vol. 13, 1996, pp. 478-496.

[PI2] J.K. Pinto, and P. Rouhiainen, “Developing a customer-based project success measurement”, *Proceedings of the 14<sup>th</sup> World Congress on Project Management*, Ljubljana, Slovenia, 1998, pp. 829-835.

[GL] Robert L. Glass, “Evolving a New Theory of Project Success”, *Communications of the ACM*, Vol.42, No. 11, November 1999, pp. 17-19.

[LI] Linberg, Kurt R. “Software Developer Perceptions about Software Project Failure: A Study”. *The Journal of Systems and Software*, Vol.49, Number 2/3, (December 30, 1999), pp. 177-192.

[PR1] J. D. Procaccino, J. M. Verner, K. M. Shelfer and D. Gefen, "What Do Software Practitioners Really Think About Project Success: An Exploratory Study", *Journal of Systems and Software*, Vol. 78, 2005. pp 194-203.

[PR2] J. D. Procaccino, "Quantitative Models for Early Prediction of Software Development Success: A Practitioner's Perspective", Doctoral Dissertation, Drexel University, October 2002.

[PR3] J. D. Procaccino, J. M. Verner, M. E. Darter, W. J. Amadio, “Toward predicting software development success from the perspective of practitioners: an explanatory Bayesian model”, *Journal of Information Technology*, Vol. 20, 2005, pp. 187-200.

[CO1] T. J. Cooke-Davies, “Consistently Doing the Right Projects and Doing Them Right – What Metrics Do You Need?”, *2004 PMI Global Congress Proceedings*, Prague, Czech Republic, 2004.

[CO2] T. J. Cooke-Davies, “The “real” success factors on projects”, *Int. Journal of Project Management*, Vol. 20, 2002, pp. 185-190.

[MU] A. K. Munns, and B. F. Bjeirmi, “The role of project management in achieving project success”, *Int. Journal of Project Management*, Vol. 12, No. 2, 1996, pp. 81-87.

[VE] Verner, J.M.; Evanco, W.M., “In-house software development: What project management practices lead to success”, *IEEE Software*, Vol. 22, Issue 1, Jan.-Feb. 2005, pp. 86–93.

[WO] C. Wohlin, and A. von Mayrhauser, “Assessing Project Success using Subjective Evaluation Factors”, *Software Quality Control*, Vol. 9, Issue 1, January 2001, pp 43-70.

[OS] Osmundson, J. S., Michael, J.B., Machniak, M.J., Grossman, M.A., “Quality Management Metrics for Software Development”, *Information and Management*, Vol. 40, 2003, pp. 799-812.

[WH] A. S. White, “External disturbance control for software project management”, *Int. Journal of Project Management*, Vol. 24, 2006, pp. 127-135.

[BR] B. B. Brown, *Employees’ Organizational Commitment and Their Perception of Supervisors’ Relations-Oriented and Task-Oriented Leadership Behaviors*, Dissertation for Doctor of Philosophy in Human Development, Virginia Polytechnic Institute and State University, Falls Church, Virginia, March 25, 2003.

[BA] S. G. Baugh, and R. M. Roberts, “Professional and Organizational Commitment among Engineers: Conflicting or Complementing”, *IEEE Transactions on Engineering Management*, Vol. 41, No. 2, May 1994.

[PA] A. K. Paul, and R. N. Anantharaman, “Impact of people management practices on organizational performance: analysis of a casual model”, *Int. Journal of Human Resource Management*, Vol. 14, No. 7, November 2003, pp. 1246-1266.

[KI] D. C. Kinlaw, *Superior Teams: What they are and how to develop them*, Gower Publishing Ltd., Hampshire, England, 1998.

[MU] R. Muller, “Determinants for external communications of IT project managers”, *Int. Journal of Project Management*, Vol. 21, 2003, pp. 345-354.

[BA] V.R. Basili, and H. D. Rombach, “The TAME Project: Towards Improvement-Oriented Software Environments”, *IEEE Trans. On Software Engineering*, Vol. SE-14, No. 6, June 1988, pp. 758-773.



[CL] B. K. Clark, “Quantifying the effects of process improvement on effort”, *IEEE Software*, November/December 2000, pp. 65-70.

[MN] Machniak, M.J., *Development of a quality management metric (QMM) measuring software program management quality*, Master’s thesis, Naval Postgraduate School, Monterey, CA, December 1999.

[FE] Norman Fenton, and Shari Lawrence Pfleeger, *Software Metrics: A Rigorous and Practical Approach*, Revised Printing, PWS Publishing Company, MA, 1997.

[ER] H. Erdogmus, “Cost-Effectiveness indicator for software development”, *Proceedings of the International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, Madrid, Spain. September 20-21, 2007.

[DM] K. A. Demir, and John S. Osmundson, “A Theory of Software Project Management and PROMOL: A Project Management Modeling Language”, Technical Report, NPS-IS-08-006, Naval Postgraduate School, Monterey, CA, USA, March 2008.

[PI3] J.K. Pinto, *Project Management Handbook*, Chapter 1, Project Management Institute, Jossey-Bass Inc., San Francisco, California, 1998, pp. 11.

[JA] A. Jaafari, “Modeling of Large Projects”, Chapter 13 in *The Wiley Guide to Managing Projects*, Edited by P. W.G. Morris, and J. K. Pinto, John Wiley and Sons, Inc., Hoboken, New Jersey, 2004, pp. 301-302.

[CO3] T. Cooke-Davies, “Project Success”, Chapter 5 in *The Wiley Guide to Managing Projects*, Edited by P. W.G. Morris, and J. K. Pinto, John Wiley and Sons, Inc., Hoboken, New Jersey, 2004, pp. 99-122.

[EV] M. E. Evans, A. M. Abela, and T. Beltz, “Seven Characteristics of Dysfunctional Software Projects”, *Crosstalk*, Vol. 15, No. 4, April 2002, pp. 16-20.

[FU] A. Fuggetta, “Software Process: A Roadmap”, *Proceeding of the Conference on the Future of Software Engineering*, Limerick, Ireland, 2000, pp. 25-34

[GR1] J. Lee, M. Gruninger, Y. Jin, T. Malone, A. Tate, G. Gost and other members of the PIF Working Group, *The PIF Process Interchange Format and Framework Version 1.1*, May 1996.

[GR2] J. Lee, M. Gruninger, Y. Jin, T. Malone, A. Tate, G. Gost and other members of the PIF Working Group, “The PIF Process Interchange Format and Framework Version 1.2”, *The Knowledge Engineering Review*, Vol. 13, No. 1, pp. 91-120, Cambridge University Press, March 1998.

[SC] C. Schlenoff, A. Knutilla, S. Ray, “A Robust Process Ontology for Manufacturing Systems Integration”, *Proc. of the 2<sup>nd</sup> Int. Conference on Engineering Design and Automation*, Maui, Hawaii, August 7-14, 1998.

[KR] P. Kruchten, “Unified Process Model (UMP)- A Model of the Rational Unified Process”, *Proc. of the International Process Technology Workshop*, Villard de Lans, France, September 1999.

[PE] A. Pease, Core Plan Representation, version 4, November 1998.

[WfMC] Workflow Management Coalition, Interface 1: Process Definition Interchange Process Model, WfMC TC-1016-P, November 1998.

[SCH] A.-W. Scheer, ARIS-Business Process Frameworks, 3<sup>rd</sup> Edition, Springer-Verlag, Berlin, 1999.

[BR] E. Breton, and J. Bezivin, “An Overview of Industrial Process Meta-Models”, *13<sup>th</sup> International Conference on Software & Systems Engineering and their Applications*, December 5-8, Paris, France, 2000.

[BMM] Object Management Group, Business Motivation Model Specification, version 1.0, September 2007.

[BPDM] Object Management Group, Business Process Definition Metamodel, Beta 1, July 2007.

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. Kadir Alpaslan Demir  
Department of Computer Science  
Naval Postgraduate School  
Monterey, California
4. Dr. Bret J. Michael  
Department of Computer Science  
Naval Postgraduate School  
Monterey, California
5. Dr. John S. Osmundson  
Department of Information Sciences  
Naval Postgraduate School  
Monterey, California
6. Deniz Kuvvetleri Komutanligi (Turkish Navy Headquarters)  
Bakanliklar, 06100, Ankara, Turkey
7. Deniz Harp Okulu Komutanligi (Turkish Naval Academy)  
Tuzla, 34942, Istanbul, Turkey
8. Arastirma Merkezi Komutanligi  
Yazilim Gelistirme Grup Baskanligi  
(Turkish Navy Software Development Center)  
Ankara Cad., No: 252, P.K. 46, Pendik, 34890, Istanbul, Turkey
9. Deniz Bilimleri Enstitusu ve Muhendisligi Mudurlugu  
(Turkish Naval Science Institution and Engineering School)  
Tuzla, 34942, Istanbul, Turkey
10. Daniel Port, Assistant Professor of Information Technology Management  
Department of Information Technology Management  
2044 Maile Way, E-601J  
Honolulu, HI, 96822, USA